

4 / PRTS

10/507024
DT15 Rec'd PCT/PTO 08 SEP 2004

WO 03/077124

PCT/US03/06853

INTERACTION DESIGN SYSTEM

Related Applications

This application claims the benefit of U.S. Provisional Application 60/362,507 filed March 7, 2002.

5 Technical Field of the Invention

The present invention relates to an interactive system that is useful in aiding a designer to design and generate user interfaces. For example, the interactive system is useful in aiding the designer to design and 10 generate user interfaces for multiple devices (e.g., cellular telephones, internet browsers, personal digital assistants).

Background of the Invention

15 Many tools have been created to aid a human in the design of products. For example, computer aided design (CAD), computer aided engineering (CAE), and computer aided manufacturing (CAM) systems have been developed to assist humans in the design and manufacture 20 of various products. These tools, however, are not easily adaptable to a changing environment. That is, changing technologies relative to the products that can be designed by these tools soon make these tools obsolete. These tools also are not flexible. That is,

each of these tools is typically limited to a specific domain to which it applies and does not accommodate other domains. For example, many CAD systems that aid a designer in the design of integrated circuits do not also 5 aid a designer in the design of automobile engines.

Moreover, these tools provide at most only a limited capability of designing an interface between the product or system being designed and the user of that product or system.

10 Accordingly, there is an interest in developing a design tool that can assist humans in the design of user interfaces in a more complex environment. Such a design tool should not be limited in the domains to which it applies or in the user interfaces that can be provided 15 as an output of the design tool. Thus, the design tool, for example, should permit the design of user interfaces based on (i) the domains within which the user interface is to be used, (ii) the tasks that users will want to perform with respect to the user interface, (iii) the 20 interaction delivery devices that can be used for the delivery of the user interface to the users, (iv) the roles, preferences, and limitations of the user or users who are to use the user interfaces, and/or (v) the presentation elements (i.e., display objects) to be used

to display input/output information to the user or users during use of the user interface being designed.

Existing design tools do not integrate these domain, task, available interaction delivery device, 5 user, and presentation element considerations into a flexible interactive design tool that provides a comprehensive approach to information presentation and interaction, that enables the designer to quickly obtain and process information in a wide variety of environments 10 and across a wide variety of tasks, that enables a designer to select from a variety of interaction delivery devices to accomplish tasks, that increases consistency and accuracy of the integration and dissemination of information by the use of common models, that 15 accommodates changing equipment needs, and/or that provides user specific, device appropriate interactions. Moreover, existing design tools do not perform any reasoning with regard to the domain, task, interaction delivery device, user, and/or the presentation element 20 considerations discussed above.

The present invention overcomes one or more of these or other problems.

Summary of the Invention

In accordance with one aspect of the present invention, a method of interactively designing a user interface comprises the following: receiving a domain model, a user model, a task model, and a device model,

5 wherein the domain model characterizes an application for which the user interface is to be used, wherein the user model characterizes users who are to interface with the user interface, wherein the task model characterizes tasks to be performed between the user interface and the

10 users, and wherein the device model characterizes interaction delivery devices that are available to deliver the user interface; and, matching characteristics in the domain model, the user model, the task model, and the device model so as to construct the

15 user interface.

In accordance with another aspect of the present invention, a method of interactively designing a user interface comprises the following: creating a domain model, wherein the domain model contains information characterizing a designer selected

20 application in a designer selected domain; creating a user model, wherein the user model contains information characterizing users of the user interface; creating a task model, wherein the task model contains task

primitives to be performed between the user and the user interface, and wherein the task model also contains types of information required by the task primitives; creating a device model, wherein the device model contains

5 information characterizing interaction delivery devices that are available to deliver the user interface; and, matching the information contained in the domain model, the user model, and the task model to the information contained in the device model and to presentation

10 elements contained in a presentation elements so as to construct the user interface, wherein the presentation elements comprise objects of the user interface that present information to the user.

In accordance with still another aspect of the

15 present invention, a method of interactively designing a user interface comprises the following: storing a domain model in a computer readable memory, wherein the domain model contains information characterizing data, concepts, and relations of an application in a domain as specified

20 by a designer; storing a user model in the computer readable memory, wherein the user model contains information characterizing roles and preferences of users of the user interface; storing a task model in the computer readable memory, wherein the task model contains

task primitives to be performed between the user and the user interface, information required of the task primitives, and sequences of the task primitives; storing a device model in the computer readable memory, 5 wherein the device model contains information including modality characterizing interaction delivery devices that are available to deliver the user interface; matching the interaction delivery devices in the device model to the information required of the task primitives and to 10 the information characterizing the users so as to identify interaction delivery devices that support the information requirements and the users; matching presentation elements to the task primitives and to the data, concepts, and relations of the domain model so as 15 to identify ones of the presentation elements that support the task primitives and the data, concepts, and relations of the domain model; and, generating the user interface based on the identified interaction delivery device and the identified presentation elements.

20 In accordance with yet another aspect of the present invention, a method of interactively designing a system comprises the following: storing a domain model, a user model, a task model, and a device model in a computer readable memory, wherein the domain model

characterizes an application for which the system is to be used, wherein the user model characterizes a user who is to use the system, wherein the task model characterizes tasks to be performed between the system and the user, and wherein the device model characterizes devices to support the system; and, matching characteristics in the domain model, the user model, the task model, and the device model so as to construct the system.

10

Brief Description of the Drawings

These and other features and advantages will become more apparent from a detailed consideration of the invention when taken in conjunction with the drawings in which:

Figure 1 illustrates a computer useful in implementing an interaction design system according to an embodiment of the present invention;

Figure 2 illustrates the architecture of the interaction design system according to an embodiment of the present invention; and,

Figures 3A and 3B illustrate a flow chart of a program that can be used for the reasoning engine of Figure 2.

Detailed Description

Figure 1 illustrates a computer 10 that can be used to implement an interaction design system 12 according to one embodiment of the present invention. As shown in Figure 1, the computer 10 includes one or more input devices 14 such as a keyboard, a mouse, and/or a modem that can be used by a designer to provide the various inputs required by the interaction design system 12 for the design and generation of user interfaces. For example, the designer can use a keyboard of the input devices 14 in providing these inputs to the interaction design system 12 residing on the computer 10.

Alternatively, one or more of these inputs can be generated remotely and supplied to the interaction design system 12 residing on the computer 10 through a modem of the input devices 14. As a further alternative, the designer can create the models and/or library discussed below on any suitable machine, save the models and/or library that the designer has created on a memory device, and load the contents of the memory device into the computer 10 at the appropriate time.

The computer 10 includes one or more output devices 16 such as a printer, a display, and a modem that

can be used by the designer in interacting with the interaction design system 12 executing on the computer 10. A common modem may be used as one of the input devices 14 and one of the output devices 16. The user 5 interfaces designed and generated by the interaction design system 12 executing on the computer 10 can be provided to the designer as files in XML that the designer or others can then load on the interaction delivery devices selected by the interaction design 10 system 12 to deliver (display visually, audibly, and/or otherwise) the user interfaces to the users.

The computer 10 also includes a memory 18 which may be in the form of random access memory, such as a floppy disk drive and/or a hard disk drive, and read only 15 memory. The memory 18 stores the interaction design system 12 that is executed by the computer 10 to design and generate user interfaces, and may be used by the interaction design system 12 during its execution. The memory 18 may further be used to store the various inputs 20 (models and library) that may be created by the designer and that are used by the interaction design system 12 during its execution to design and generate user interfaces.

Finally, the computer 10 includes a processor 20 that executes the interaction design system 12 stored by the memory 18.

Figure 2 illustrates the architecture of the 5 interaction design system 12. The interaction design system 12 includes various inputs that the interaction design system 12 uses in designing and generating user interfaces. These inputs include a domain model 22, user models 24, task models 26, and device models 28. In 10 addition, the interaction design system 12 includes a presentation elements library 30 that stores various presentation elements (objects) and a set of characteristics for each presentation element. The characteristics for each presentation element can be 15 matched to the inputs created by the designer to allow a reasoning engine 32 of the interaction design system 12 to make qualitative judgments about the ability of the corresponding presentation elements of the presentation elements library 30 and the interaction delivery devices 20 of the device models 28 to support the performance of the various interactions required of the user interfaces and the input and output of the information required by these interactions.

Thus, the reasoning engine 32 of the interaction design system 12 makes qualitative judgments about the ability of the presentation elements stored in the presentation element library 30 and the interaction 5 delivery devices of the device models 28 (i) to support the application and domain specified by the designer in the domain model 22, (ii) to interact with the users as defined by the designer in the user models 24, (iii) to support the task primitives (i.e., the actions performed 10 between the user interface and the users) as specified by the designer in the task models 26, and (iv) the information required by the task primitives as specified by the designer in the task models 26.

As indicated above, the designer creates the 15 domain model 22 as an input of the interaction design system 12. More specifically, the designer creates the domain model 22 as a machine interpretable domain-specific representation of the relevant domain attributes to be given to a user interface. These attributes 20 include the data, information, concepts, and/or relations pertinent to the application and domain for which the user interface is being designed. The form of the domain representation should be consistent with the other models that are created as inputs so that the interaction design

system 12 can properly match the characteristics and specifications of the interaction delivery devices of the device models 28 and of the presentation elements of the presentation elements library 30 with the characteristics and specifications provided in the domain model 22, the user models 24, and the task models 26. The domain model 22 should be application and domain specific.

In creating the domain model 22, the designer will already have in mind a particular application in a particular domain. For example, the designer may wish to design and generate user interfaces for the application of prescription drug ordering and filling in the domain of medicine. The designer must then model the data, information, concepts, and/or relations for this application. For example, in the application of prescription drug ordering and filling, the designer may determine that the user interfaces will deal with various items of information such as doctors, patients, pharmacists, medications, amounts, length of character strings required for the display of each of these entities, etc., with the relationships between these various entities.

The domain model 22 may be hierarchical having, for example, three levels in the hierarchy with domain

being the top level, domain elements being the next level down, and domain data being the bottom level. In the prescription drug ordering and filling example, the domain is prescription drug ordering and filling, the 5 domain elements are doctors, patients, care givers, etc, and the domain data are amounts/values, labels/names, times such as pick-up times, lengths of character strings, etc.

In developing the domain model 22, the 10 designer, for example, develops a meta-ontology that specifies the general required structure and vocabulary of the knowledge characterizing the designated domain, designs a specific schema for the selected application of the user interfaces being designed, and then populates 15 the schema with the specific information, relationships; and concepts pertinent to this schema. The Resource Description Framework (RDF) notation may be used to create the schema and to populate it, although any other schema specific notation may be used for these purposes. 20 Appendix A discloses an exemplary class hierarchy that may structure a domain-independent architecture for such a framework.

The RDF Specification is described in the documents "Resource Description Framework (RDF) Model and Syntax

Specification: W3C Recommendation 22 February 1999", and "RDF/XML Syntax Specification (Revised):W3C Working Draft 23 January 2003", both herein incorporated by reference. The domain model 22 created by the designer is stored in 5 the memory 18 so that it is available to the interaction design system 12 during its execution on the computer 10.

The following is an abbreviated example of an RDF schema for the domain model 22 in the prescription drug ordering and filling application:

10

```
<rdfs:Class rdf:about = "Prescription"/>
<rdfs:Class rdf:about = "Medication"/>
<rdfs:Class rdf:about = "Pharmacy"/>
<rdf:Property rdf:about = "specifiesMedication"/>
15      <rdfs:domain rdf:resource = "Prescription"/>
          <rdfs:range rdf:about = "Medication"/>
<rdf:Property>
<rdf:Property rdf:about = "soldBy"/>
          <rdfs:domain rdf:resource = "Medication"/>
20      <rdfs:range rdf:resource = "Pharmacy"/>
<rdf:Property>
```

As can be seen, this RDF schema is only a partial schema for the prescription drug ordering and filling

application and should be expanded to include the other domain elements and domain data for a practical domain model.

In the user models 24 created by the designer, 5 the designer captures the preferences, roles, and abilities (or limitations) of the users who are identified by the designer as the users of the user interfaces being designed. The preferences, roles, and abilities of the users can be captured using a flexible 10 notation such as RDF. The preferences, roles, and abilities of the users include, for example, role descriptions, interaction delivery device preferences, modality (such as visual or audible) preferences, physical challenges that may confront the users of the 15 user interface being designed, etc. Accordingly, the designer has the responsibility to understand the users and the application requirements so that the designer can create the user models 24 so that they define the relationships between the users and the application.

20 In the prescription drug ordering and filling example above, the users of the prescription drug ordering and filling interfaces may be any one or more of the following: doctors, patients, care givers, those who may be designated by the patients to pick up the

prescribed medicines, etc. The designer should keep in mind the preferences and abilities as well as the roles of all of these people in creating the user models 24. The user models 24 created by the designer are stored in 5 the memory 18 so that they are available to the interaction design system 12 during its execution on the computer 10.

The following is an abbreviated example of an RDF schema for the user models 24 in the prescription 10 drug ordering and filling application:

```
<rdfs:Class rdf:about="Person"/>

<rdfs:Class rdf:about="PhysicalAbilities"/>

<rdfs:Class rdf:about="VisualAcuity">
15      <rdfs:subClassOf rdf:resource="PhysicalAbilities"/>
</rdfs:Class>

<rdf:Property rdf:about="VisualRating">
    <rdfs:domain rdf:resource="VisualAcuity"/>
    <rdfs:range rdf:resource="Literal"/>
20      <allowedValues>0</allowedValues>
      <allowedValues>1</allowedValues>
      <allowedValues>2</allowedValues>
```

```
<rdf:Property rdf:about="vision">  
    <rdfs:domain rdf:resource="Person"/>  
    <rdfs:range rdf:resource="VisualAcuity"/>  
5    </rdf:Property>
```

As can be seen, this RDF schema is only a partial schema for the prescription drug ordering and filling application and should be expanded to include other 10 preferences, roles, and abilities of the users who are to use the interfaces being designed.

In creating the task models 26, the designer captures the actions to be performed by the users when using the user interfaces being designed, the goals to be 15 achieved by the user when using the user interfaces being designed, and the information required to perform the actions and achieve the goals. The task models 26 are meant to capture what actions on the part of the user the interfaces are intended to afford or support. The tasks 20 can be captured using a flexible notation such as RDF that includes order-of-flow, nouns (i.e., the information required by the tasks), and the tasks that are to be performed. Accordingly, the designer has the responsibility to understand the task requirements within

the application and to apply the modeling notation to capture the specific task requirements.

In task modeling, the designer decomposes each task or interaction to be performed by the user into task primitives, an order to flow from each task primitive, and the type of information required by the task primitive. The task primitives may include any actions that a designer is likely to require between users and user interfaces for the relevant application in the relevant domain. For example, task primitives may include receive, instantiate, compare, monitor, assert, select, control, retract, change, detect, direct attention, navigate, adjust, etc. View, listen, review, and assess may be synonyms of the task primitive receive.

Configure may be a synonym of instantiate. Observe and watch may be synonyms of the task primitive monitor. Set, enter, input, record, and declare may be synonyms of the task primitive assert. Pick and select-item-from-set may be synonyms of the task primitive select. Maintain, direct, and command may be synonyms of the task primitive control. Undo and withdraw may be synonyms of the task primitive retract. Modify, update, edit, and alter may be synonyms of the task primitive change. Identify, catch, and notice may be synonyms of the task primitive

detect. Focus may be a synonym of the task primitive direct attention. Move may be a synonym of the task primitive navigate. Configure may be a synonym of the task primitive adjust.

5 Also, as indicated above, the designer captures order-of-flow. Order-of-flow refers to the precedence between task primitives, and can be designated by junctions and links in the notation applied to the task models 26. Such junctions may include, for example, AND,
10 OR, and XOR junctions that may be synchronous or asynchronous. Links, for example, may be used to indicate that action B does not start before action A completes, that action A must be followed by action B, that action B must be preceded by action A, that actions
15 A and B are both required, etc.

Moreover, the task models 26 also include the information required by the task primitives of the tasks being modeled. For example, the task primitive receive requires that information be received by the user. The
20 designer defines the type of information for this task primitive in the task models 26. As another example, the task primitive instantiate requires that information be instantiated. The designer defines the type of information for this task primitive instantiation in the

task models 26. As still another example, the task primitive compare requires that certain information be compared to certain other information. The designer defines the type of information for this task primitive 5 compare in the task models 26. The other task primitives also require information typing, and the designer defines the type of information for each of these task primitives that is used in the task models 26 as well. The task models 26 created by the designer are stored in the 10 memory 18 so that they are available to the interaction design system 12 during its execution on the computer 10.

In the prescription drug ordering and filling example above, the tasks to be performed may include 15 directing the user to choose from among a displayed set of pharmacies, to choose whether the prescription is a refill, to designate a person to pick up the medication, to indicate that the prescription is to be moved from one pharmacy to another, etc.

20 The following is an abbreviated example of an RDF document adhering to the IDS RDF Schema for the task modeling 26 in the prescription drug ordering and filling application:

```
<userTask rdf:about="idsTask_00025"
  identifier="tskFillPrescription"
  name="Fill Prescription"
  primitive="NONPRIMITIVE"
  5   rdfs:label="tskFillPrescription">
    <tasks rdf:resource="idsTask_00026" />
    <entryTask rdf:resource="idsTask_00026" />
    <tasks rdf:resource="idsTask_00027" />
    <tasks rdf:resource="idsTask_00028" />
    10  <exitTask rdf:resource="idsTask_00029" />
    <tasks rdf:resource="idsTask_00029" />
    <tasks rdf:resource="idsTask_00032" />
    <matchRequirements rdf:resource="idsTask_00036" />
  </userTask>
  15  <junction rdf:about="idsTask_00026"
    identifier="AND1"
    operator="AND"
    primitive="JUNCTION"
    rdfs:label="AND1">
    <parent rdf:resource="idsTask_00025" />
    <mate rdf:resource="idsTask_00029" />
    <followingRelations rdf:resource="idsTask_00030" />
    <followingRelations rdf:resource="idsTask_00034" />
  </junction>
```

```
<userTask rdf:about="idsTask_00027"
  1   classElement="medication"
  2   identifier="tskSelectMedication"
  3   name="Select Medication"
  4
  5   primitive="SELECT"
  6
  7   rdfs:label="tskSelectMedication">
  8
  9   <parent rdf:resource="idsTask_00025"/>
 10
 11   <precedingRelations rdf:resource="idsTask_00030"/>
 12
 13   <followingRelations rdf:resource="idsTask_00031"/>
 14
 15   <matchRequirements rdf:resource="idsTask_00036"/>
 16
 17   </userTask>
 18
 19
 20
 21
 22
 23
 24
 25
 26
 27
 28
 29
 30
 31
 32
 33
 34
 35
 36
 37
 38
 39
 40
 41
 42
 43
 44
 45
 46
 47
 48
 49
 50
 51
 52
 53
 54
 55
 56
 57
 58
 59
 60
 61
 62
 63
 64
 65
 66
 67
 68
 69
 70
 71
 72
 73
 74
 75
 76
 77
 78
 79
 80
 81
 82
 83
 84
 85
 86
 87
 88
 89
 90
 91
 92
 93
 94
 95
 96
 97
 98
 99
 100
 101
 102
 103
 104
 105
 106
 107
 108
 109
 110
 111
 112
 113
 114
 115
 116
 117
 118
 119
 120
 121
 122
 123
 124
 125
 126
 127
 128
 129
 130
 131
 132
 133
 134
 135
 136
 137
 138
 139
 140
 141
 142
 143
 144
 145
 146
 147
 148
 149
 150
 151
 152
 153
 154
 155
 156
 157
 158
 159
 160
 161
 162
 163
 164
 165
 166
 167
 168
 169
 170
 171
 172
 173
 174
 175
 176
 177
 178
 179
 180
 181
 182
 183
 184
 185
 186
 187
 188
 189
 190
 191
 192
 193
 194
 195
 196
 197
 198
 199
 200
 201
 202
 203
 204
 205
 206
 207
 208
 209
 210
 211
 212
 213
 214
 215
 216
 217
 218
 219
 220
 221
 222
 223
 224
 225
 226
 227
 228
 229
 230
 231
 232
 233
 234
 235
 236
 237
 238
 239
 240
 241
 242
 243
 244
 245
 246
 247
 248
 249
 250
 251
 252
 253
 254
 255
 256
 257
 258
 259
 260
 261
 262
 263
 264
 265
 266
 267
 268
 269
 270
 271
 272
 273
 274
 275
 276
 277
 278
 279
 280
 281
 282
 283
 284
 285
 286
 287
 288
 289
 290
 291
 292
 293
 294
 295
 296
 297
 298
 299
 300
 301
 302
 303
 304
 305
 306
 307
 308
 309
 310
 311
 312
 313
 314
 315
 316
 317
 318
 319
 320
 321
 322
 323
 324
 325
 326
 327
 328
 329
 330
 331
 332
 333
 334
 335
 336
 337
 338
 339
 340
 341
 342
 343
 344
 345
 346
 347
 348
 349
 350
 351
 352
 353
 354
 355
 356
 357
 358
 359
 360
 361
 362
 363
 364
 365
 366
 367
 368
 369
 370
 371
 372
 373
 374
 375
 376
 377
 378
 379
 380
 381
 382
 383
 384
 385
 386
 387
 388
 389
 390
 391
 392
 393
 394
 395
 396
 397
 398
 399
 400
 401
 402
 403
 404
 405
 406
 407
 408
 409
 410
 411
 412
 413
 414
 415
 416
 417
 418
 419
 420
 421
 422
 423
 424
 425
 426
 427
 428
 429
 430
 431
 432
 433
 434
 435
 436
 437
 438
 439
 440
 441
 442
 443
 444
 445
 446
 447
 448
 449
 450
 451
 452
 453
 454
 455
 456
 457
 458
 459
 460
 461
 462
 463
 464
 465
 466
 467
 468
 469
 470
 471
 472
 473
 474
 475
 476
 477
 478
 479
 480
 481
 482
 483
 484
 485
 486
 487
 488
 489
 490
 491
 492
 493
 494
 495
 496
 497
 498
 499
 500
 501
 502
 503
 504
 505
 506
 507
 508
 509
 510
 511
 512
 513
 514
 515
 516
 517
 518
 519
 520
 521
 522
 523
 524
 525
 526
 527
 528
 529
 530
 531
 532
 533
 534
 535
 536
 537
 538
 539
 540
 541
 542
 543
 544
 545
 546
 547
 548
 549
 550
 551
 552
 553
 554
 555
 556
 557
 558
 559
 559
 560
 561
 562
 563
 564
 565
 566
 567
 568
 569
 570
 571
 572
 573
 574
 575
 576
 577
 578
 579
 580
 581
 582
 583
 584
 585
 586
 587
 588
 589
 589
 590
 591
 592
 593
 594
 595
 596
 597
 598
 599
 599
 600
 601
 602
 603
 604
 605
 606
 607
 608
 609
 609
 610
 611
 612
 613
 614
 615
 616
 617
 618
 619
 619
 620
 621
 622
 623
 624
 625
 626
 627
 628
 629
 629
 630
 631
 632
 633
 634
 635
 636
 637
 638
 639
 639
 640
 641
 642
 643
 644
 645
 646
 647
 648
 649
 649
 650
 651
 652
 653
 654
 655
 656
 657
 658
 659
 659
 660
 661
 662
 663
 664
 665
 666
 667
 668
 669
 669
 670
 671
 672
 673
 674
 675
 676
 677
 678
 679
 679
 680
 681
 682
 683
 684
 685
 686
 687
 688
 689
 689
 690
 691
 692
 693
 694
 695
 696
 697
 698
 699
 699
 700
 701
 702
 703
 704
 705
 706
 707
 708
 709
 709
 710
 711
 712
 713
 714
 715
 716
 717
 718
 719
 719
 720
 721
 722
 723
 724
 725
 726
 727
 728
 729
 729
 730
 731
 732
 733
 734
 735
 736
 737
 738
 739
 739
 740
 741
 742
 743
 744
 745
 746
 747
 748
 749
 749
 750
 751
 752
 753
 754
 755
 756
 757
 758
 759
 759
 760
 761
 762
 763
 764
 765
 766
 767
 768
 769
 769
 770
 771
 772
 773
 774
 775
 776
 777
 778
 779
 779
 780
 781
 782
 783
 784
 785
 786
 787
 788
 789
 789
 790
 791
 792
 793
 794
 795
 796
 797
 798
 799
 799
 800
 801
 802
 803
 804
 805
 806
 807
 808
 809
 809
 810
 811
 812
 813
 814
 815
 816
 817
 818
 819
 819
 820
 821
 822
 823
 824
 825
 826
 827
 828
 829
 829
 830
 831
 832
 833
 834
 835
 836
 837
 838
 839
 839
 840
 841
 842
 843
 844
 845
 846
 847
 848
 849
 849
 850
 851
 852
 853
 854
 855
 856
 857
 858
 859
 859
 860
 861
 862
 863
 864
 865
 866
 867
 868
 869
 869
 870
 871
 872
 873
 874
 875
 876
 877
 878
 879
 879
 880
 881
 882
 883
 884
 885
 886
 887
 888
 889
 889
 890
 891
 892
 893
 894
 895
 896
 897
 898
 899
 899
 900
 901
 902
 903
 904
 905
 906
 907
 908
 909
 909
 910
 911
 912
 913
 914
 915
 916
 917
 918
 919
 919
 920
 921
 922
 923
 924
 925
 926
 927
 928
 929
 929
 930
 931
 932
 933
 934
 935
 936
 937
 938
 939
 939
 940
 941
 942
 943
 944
 945
 946
 947
 948
 949
 949
 950
 951
 952
 953
 954
 955
 956
 957
 958
 959
 959
 960
 961
 962
 963
 964
 965
 966
 967
 968
 969
 969
 970
 971
 972
 973
 974
 975
 976
 977
 978
 979
 979
 980
 981
 982
 983
 984
 985
 986
 987
 988
 989
 989
 990
 991
 992
 993
 994
 995
 996
 997
 998
 999
 999
 1000
 1001
 1002
 1003
 1004
 1005
 1006
 1007
 1008
 1009
 1009
 1010
 1011
 1012
 1013
 1014
 1015
 1016
 1017
 1018
 1019
 1019
 1020
 1021
 1022
 1023
 1024
 1025
 1026
 1027
 1028
 1029
 1029
 1030
 1031
 1032
 1033
 1034
 1035
 1036
 1037
 1038
 1039
 1039
 1040
 1041
 1042
 1043
 1044
 1045
 1046
 1047
 1048
 1049
 1049
 1050
 1051
 1052
 1053
 1054
 1055
 1056
 1057
 1058
 1059
 1059
 1060
 1061
 1062
 1063
 1064
 1065
 1066
 1067
 1068
 1069
 1069
 1070
 1071
 1072
 1073
 1074
 1075
 1076
 1077
 1078
 1079
 1079
 1080
 1081
 1082
 1083
 1084
 1085
 1086
 1087
 1088
 1089
 1089
 1090
 1091
 1092
 1093
 1094
 1095
 1096
 1097
 1098
 1099
 1099
 1100
 1101
 1102
 1103
 1104
 1105
 1106
 1107
 1108
 1109
 1109
 1110
 1111
 1112
 1113
 1114
 1115
 1116
 1117
 1118
 1119
 1119
 1120
 1121
 1122
 1123
 1124
 1125
 1126
 1127
 1128
 1129
 1129
 1130
 1131
 1132
 1133
 1134
 1135
 1136
 1137
 1138
 1139
 1139
 1140
 1141
 1142
 1143
 1144
 1145
 1146
 1147
 1148
 1149
 1149
 1150
 1151
 1152
 1153
 1154
 1155
 1156
 1157
 1158
 1159
 1159
 1160
 1161
 1162
 1163
 1164
 1165
 1166
 1167
 1168
 1169
 1169
 1170
 1171
 1172
 1173
 1174
 1175
 1176
 1177
 1178
 1179
 1179
 1180
 1181
 1182
 1183
 1184
 1185
 1186
 1187
 1188
 1189
 1189
 1190
 1191
 1192
 1193
 1194
 1195
 1196
 1197
 1198
 1198
 1199
 1199
 1200
 1201
 1202
 1203
 1204
 1205
 1206
 1207
 1208
 1209
 1209
 1210
 1211
 1212
 1213
 1214
 1215
 1216
 1217
 1218
 1219
 1219
 1220
 1221
 1222
 1223
 1224
 1225
 1226
 1227
 1228
 1229
 1229
 1230
 1231
 1232
 1233
 1234
 1235
 1236
 1237
 1238
 1239
 1239
 1240
 1241
 1242
 1243
 1244
 1245
 1246
 1247
 1248
 1249
 1249
 1250
 1251
 1252
 1253
 1254
 1255
 1256
 1257
 1258
 1259
 1259
 1260
 1261
 1262
 1263
 1264
 1265
 1266
 1267
 1268
 1269
 1269
 1270
 1271
 1272
 1273
 1274
 1275
 1276
 1277
 1278
 1279
 1279
 1280
 1281
 1282
 1283
 1284
 1285
 1286
 1287
 1288
 1289
 1289
 1290
 1291
 1292
 1293
 1294
 1295
 1296
 1297
 1298
 1298
 1299
 1299
 1300
 1301
 1302
 1303
 1304
 1305
 1306
 1307
 1308
 1309
 1309
 1310
 1311
 1312
 1313
 1314
 1315
 1316
 1317
 1318
 1319
 1319
 1320
 1321
 1322
 1323
 1324
 1325
 1326
 1327
 1328
 1329
 1329
 1330
 1331
 1332
 1333
 1334
 1335
 1336
 1337
 1338
 1339
 1339
 1340
 1341
 1342
 1343
 1344
 1345
 1346
 1347
 1348
 1349
 1349
 1350
 1351
 1352
 1353
 1354
 1355
 1356
 1357
 1358
 1359
 1359
 1360
 1361
 1362
 1363
 1364
 1365
 1366
 1367
 1368
 1369
 1369
 1370
 1371
 1372
 1373
 1374
 1375
 1376
 1377
 1378
 1379
 1379
 1380
 1381
 1382
 1383
 1384
 1385
 1386
 1387
 1388
 1389
 1389
 1390
 1391
 1392
 1393
 1394
 1395
 1396
 1397
 1398
 1398
 1399
 1399
 1400
 1401
 1402
 1403
 1404
 1405
 1406
 1407
 1408
 1409
 1409
 1410
 1411
 1412
 1413
 1414
 1415
 1416
 1417
 1418
 1419
 1419
 1420
 1421
 1422
 1423
 1424
 1425
 1426
 1427
 1428
 1429
 1429
 1430
 1431
 1432
 1433
 1434
 1435
 1436
 1437
 1438
 1439
 1439
 1440
 1441
 1442
 1443
 1444
 1445
 1446
 1447
 1448
 1449
 1449
 1450
 1451
 1452
 1453
 1454
 1455
 1456
 1457
 1458
 1459
 1459
 1460
 1461
 1462
 1463
 1464
 1465
 1466
 1467
 1468
 1469
 1469
 1470
 1471
 1472
 1473
 1474
 1475
 1476
 1477
 1478
 1479
 1479
 1480
 1481
 1482
 1483
 1484
 1485
 1486
 1487
 1488
 1489
 1489
 1490
 1491
 1492
 1493
 1494
 1495
 1496
 1497
 1498
 1498
 1499
 1499
 1500
 1501
 1502
 1503
 1504
 1505
 1506
 1507
 1508
 1509
 1509
 1510
 1511
 1512
 1513
 1514
 1515
 1516
 1517
 1518
 1519
 1519
 1520
 1521
 1522
 1523
 1524
 1525
 1526
 1527
 1528
 1529
 1529
 1530
 1531
 1532
 1533
 1534
 1535
 1536
 1537
 1538
 1539
 1539
 1540
 1541
 1542
 1543
 1544
 1545
 1546
 1547
 1548
 1549
 1549
 1550
 1551
 1552
 1553
 1554
 1555
 1556
 1557
 1558
 1559
 1559
 1560
 1561
 1562
 1563
 1564
 1565
 1566
 1567
 1568
 1569
 1569
 1570
 1571
 1572
 1573
 1574
 1575
 1576
 1577
 1578
 1579
 1579
 1580
 1581
 1582
 1583
 1584
 1585
 1586
 1587
 1588
 1589
 1589
 1590
 1591
 1592
 1593
 1594
 1595
 1596
 1597
 1598
 1598
 1599
 1599
 1600
 1601
 1602
 1603
 1604
 1605
 1606
 1607
 1608
 1609
 1609
 1610
 1611
 1612
 1613
 1614
 1615
 1616
 1617
 1618
 1619
 1619
 1620
 1621
 1622
 1623
 1624
 1625
 1626
 1627
 1628
 1629
 1629
 1630
 1631
 1632
 1633
 1634
 1635
 1636
 1637
 1638
 1639
 1639
 1640
 1641
 1642
 1643
 1644
 1645
 1646
 1647
 1648
 1649
 1649
 1650
 1651
 1652
 1653
 1654
 1655
 1656
 1657
 1658
 1659
 1659
 1660
 1661
 1662
 1663
 1664
 1665
 1666
 1667
 1668
 1669
 1669
 1670
 1671
 1672
 1673
 1674
 1675
 1676
 1677
 1678
 1679
 1679
 1680
 1681
 1682
 1683
 1684
 1685
 1686
 1687
 1688
 1689
 1689
 1690
 1691
 1692
 1693
 1694
 1695
 1696
 1697
 1698
 1698
 1699
 1699
 1700
 1701
 1702
 1703
 1704
 1705
 1706
 1707
 1708
 1709
 1709
 1710
 1711
 1712
 1713
 1714
 1715
 1716
 1717
 1718
 1719
 1719
 1720
 1721
 1722
 1723
 1724
 1725
 1726
 1727
 1728
 1729
 1729
 1730
 1731
 1732
 1733
 1734
 1735
 1736
 1737
 1738
 1739
 1739
 1740
 1741
 1742
 1743
 1744
 1745
 1746
 1747
 1748
 1749
 1749
 1750
 1751
 1752
 1753
 1754
 1755
 1756
 1757
 1758
 1759
 1759
 1760
 1761
 1762
 1763
 1764
 1765
 1766
 1767
 1768
 1769
 1769
 1770
 1771
 1772
 1773
 1774
 1775
 1776
 1777
 1778
 1779
 1779
 1780
 1781
 1782
 1783
 1784
 1785
 1786
 1787
 1788
 1789
 1789
 1790
 1791
 1792
 1793
 1794
 1795
 1796
 1797
 1798
 1798
 1799
 1799
 1800
 1801
 1802
 1803
 1804
 1805
 1806
 1807
 1808
 1809
 1809
 1810
 1811
 1812
 1813
 1814
 1815
 1816
 1817
 1818
 1819
 1819
 1820
 1821
 1822
 1823
 1824
 1825
 1826
 1827
 1828
 1829
 1829
 1830
 1831
 1832
 1833
 1834
 1835
 1836
 1837
 1838
 1839
 1839
 1840
 1841
 1842
 1843
 1844
 1845
 1846
 1847
 1848
 1849
 1849
 1850
 1851
 1852
 1853
 1854
 1855
 1856
 1857
 1858
 1859
 1859
 1860
 1861
 1862
 1863
 1864
 1865
 1866
 1867
 1868
 1869
 1869
 1870
 1871
 1872
 1873
 1874
 1875
 1876
 1877
 1878
 1879
 1879
 1880
 1881
 1882
 1883
 1884
 1885
 1886
 1887
 1888
 1889
 1889
 1890
 1891
 1892
 1893
 1894
 1895
 1896
 1897
 1898
 1898
 1899
 1899
 1900
 1901
 1902
 1903
 1904
 1905
 19
```

to be performed by the users when using the interfaces being designed.

In the device models 28, the designer captures the specifications and characteristics of the interaction 5 delivery devices that are available to deliver the user interfaces being designed when these user interfaces are invoked by the applications for which they are designed. These specifications should include the capabilities and modalities that are supported by the available 10 interaction delivery devices and that are relevant to the application. The capabilities, for example, may include bandwidth, memory, screen, lines of display, width of display, illumination, etc., and the modalities, for example, may include visual, audible, etc.

15 These specifications and characteristics of the interaction delivery devices can be captured using a flexible notation such as RDF that includes a mechanism to describe an interaction delivery device's specific input and output modalities and capabilities. The 20 designer has the responsibility to understand the interaction delivery device specification and to apply the interaction delivery device description notation to produce the device models 28. The device models 28 created by the designer are stored in the memory 18 so

that they are available to the interaction design system
12 during its execution on the computer 10.

In the prescription drug ordering and filling example above, interaction delivery devices can include
5 web browsers, personal digital assistants, telephonic interaction delivery devices, etc. These interaction delivery devices are existing interaction delivery devices that can be used to deliver the user interface to the users. In the case of audio interaction delivery
10 devices, the capabilities of such interaction delivery devices can include number of tones, word rate, speech vs. tone only, etc. In the case of visual interaction delivery devices, the capabilities of such interaction delivery devices can include number of lines, number of
15 characters per line, update rate, etc. In the case of hardware buttons, the capabilities can include yes/no/select vs. none, numeric vs. none, up/down vs. none, left/right vs. none, alphabetic vs. none, input rate, etc.

20 The following is an abbreviated example of an RDF schema for the device models 28 in the prescription drug ordering and filling application:

```
<rdfs:Class rdf:about="InteractionDevice">
```

```
        <rdfs:subClassOf rdf:resource="DomainItem" />

    </rdfs:Class>

    <rdfs:Class rdf:about="DeviceSignature">

        <rdfs:subClassOf rdf:resource="NounSignature">

    5    </rdfs:Class>

        <rdfs:Class rdf:about="AudioDisplayCharacteristics"/>

        <rdfs:Class rdf:about="HardwareButtonCharacteristics"/>

        <rdf:Property rdf:about="supportedDeviceSignature">

            <rdfs:range rdf:resource="DeviceSignature"/>

    10    <rdfs:domain rdf:resource="InteractionDevice"/>

            </rdf:Property>

            <rdf:Property rdf:about="audioDisplay">

                <rdfs:range

                    rdf:resource="AudioDisplayCharacteristics"/>

    15    <rdfs:domain rdf:resource="DeviceSignature"/>

            </rdf:Property>

            <rdf:Property rdf:about="speechOutput"

                ids:range="boolean">

                <rdfs:domain

    20    rdf:resource="AudioDisplayCharacteristics"/>

                    <rdfs:range rdf:resource="&rdfs;Literal"/>

            </rdf:Property>
```

As can be seen, this RDF schema is only a partial schema for the description of interaction devices and should be expanded to include other interaction delivery device specifications for the interaction delivery devices that 5 can potentially deliver user interfaces to users.

Presentation elements are the display objects that are used to present information to or acquire information from a user of the user interface being designed. In the context of a web browser being used as 10 the interaction delivery device, a presentation element may be a pop up menu, a pull down menu, a dialog box, a button, etc.

Each of the presentation elements stored in the presentation elements library 30 contains a set of 15 functionality and usability characteristics that the corresponding presentation element either supports or requires for correct application in a presentation. The functionality and usability characteristics describe the quality of the interaction that can be achieved given the 20 functionality and usability characteristics of the display objects. The functionality and usability characteristics for each presentation element should have a form so that they can be matched by the reasoning engine 32 to the other inputs, and so that the reasoning

engine 32 can make qualitative judgments about the ability of the presentation elements to support the input and output of the data to be exchanged between the users and the user interfaces.

5 In the prescription drug ordering and filling application, examples of presentation elements include pull down menus, dialog boxes, windows, buttons, etc.

 The following is an abbreviated example of an XML composer, written in Java, for the presentation
10 elements library 30 in the prescription drug ordering and filling application:

```
private String createXML(final Presentation
presentation) {
15         // get the task's name
        String taskName = "name=" + "\"" +
presentation.getInteractionRequirement().getTask().getName() + "\" ;
        // get the task's identifier
20         String taskIdentifier = "identifier=" + "\"" +
presentation.getInteractionRequirement().getTask().getIdentifier() + "\" ;
        // get the domain element content
```

```
        output =
    presentation.getInteractionRequirement().getDomainItem().
    getContent(presentation.getInteractionRequirement().getTa
    sk());
5      // Create outer <textbox> element tags
    String preFix = "<textBox " + taskName + " " +
    taskIdentifier + ">";
    preFix = preFix + " <label><text>" +
    presentation.getInteractionRequirement().getTask().getNam
10    e() + "</text></label>";
    //sandwich content between open and end tags
    output = preFix + output + "</textBox>";
    return output;
}
15
```

As can be seen, this XML composer is only one example and should be expanded to include other presentation elements that can potentially be used to exchange information between the users and the application for which the user

20 interfaces are being designed.

As shown in Figure 2, the domain model 22, the user models 24, the task models 26, and the device models 28 combine to define the interaction requirements of the user interface being designed. Each interaction

requirement is a combination of a task primitive and information required by the task primitive as influenced by the characteristics of the users and the application and domain for which the user interface is being 5 designed. Thus, a user interface typically involves a plurality of interaction requirements that define the totality of the way in which the users interact with the user interface being defined.

The domain model 22, the user models 24, the 10 task models 26, the device models 28, and the presentation elements library 30 are stored in the memory 18 in preparation for the execution of the reasoning engine 32. Based on the domain model 22, the user models 24, the task models 26, the device models 28, and the 15 presentation elements library 30, the reasoning engine 32 makes qualitative judgments about the best fit between the presentation elements and the interaction delivery devices in order to interact with the user through the user interface as intended by the designer.

20 A flow chart of a program that can be used for the reasoning engine 32 is shown in Figures 3A and 3B. Once the domain model 22, the user models 24, the task models 26, the device models 28 have been created and entered by the designer and stored in the memory 18, and

once the presentation elements library 30 has been populated with the presentation elements, the reasoning engine 32 may be executed.

Accordingly, a block 40 filters the interaction delivery devices of the device models 28 based on the information requirements of the task primitives defined in the tasks models 26 and the roles, preferences, and abilities of the users as defined in the user models 24 so as to form an intersection between these available interaction delivery devices, information requirements, and user characteristics. Accordingly, the information requirements of the task primitives as modeled by the designer in the tasks models 26, the roles, preferences, and abilities of the users as modeled by the designer in the user models 24, and the characteristic specifications and capabilities of the interaction delivery devices as modeled by the designer in the device models 28 are matched, and only those available interaction delivery devices that can support those information requirements and user characteristics are saved for further processing. These saved interaction delivery devices, therefore, are the filtered output of the filtering process of the block 40.

As an example, if a user is to invoke an interface to change the temperature of a room, the block 40 of the interaction design system 12 should consider those interaction delivery devices that are available to 5 change temperatures and that meet the preferences, roles and abilities of the intended users.

A block 42 filters the presentation elements stored in the presentation library 30 based on the task primitives of the task models 26 and the characteristics 10 of the information, concepts, and relations of the domain model 22 to form an intersection between the presentation elements stored in the presentation library 30, the task primitives of the task models 26, and the domain characteristics of the domain model 22. Accordingly, the 15 presentation elements, the task primitives, and the domain characteristics are matched, and only those presentation elements that can support the task primitives and domain characteristics are saved for further processing. These saved presentation elements, 20 therefore, are the output of the filtering process of the block 42.

As an example, if a task is a SELECT primitive, and the user is supposed to select a numeric value as a domain characteristic, the block 42 considers those

presentation elements that support SELECT of numeric lists.

A block 44 creates a presentation comprising each presentation element saved as a result of the 5 processing of the block 42 and the interaction delivery devices that are saved as a result of the processing of the block 40 and that support the corresponding presentation element. Accordingly, the block 44 matches the interaction delivery devices saved by the block 40 10 and the presentation elements saved by the block 42, and creates a presentation for each saved presentation element and matching interaction delivery device.

It should be understood that each presentation element is not required to support all of the interaction 15 delivery devices. Similarly, each interaction delivery device is not required to support all presentation elements. However, it is quite possible that any given presentation element saved by the block 42 will match more than one of the interaction delivery devices saved 20 by the block 40. Therefore, a presentation element may be included in more than one presentation. Each presentation comprises a presentation element and an interaction delivery device as a matching pair. The

presentations at the output of the block 44 form a fully-resolved set of usability or interaction capabilities.

A block 46 assigns a score to each presentation based on the quality of the match that produced the 5 interaction delivery devices saved by the block 40 and the quality of the match that produced the presentation elements saved by the block 42. This scoring is a qualitative judgment about how well a presentation (presentation element/interactive deliver device pair) 10 meets the domain characteristics of the domain model 22, the roles, preferences, and abilities of the users as defined in the user models 24, and the task primitives and corresponding information requirements defined in the task models 26.

15 For example, if a presentation element is to perform the task primitive receive in order to display information such as current temperature to a user, various presentation elements, such as a digital readout, a round graphic, and a thermostat graphic, can be 20 considered. Each of these presentation elements may be assigned a set of values such as DisplayScope (DS), DisplayResolution (DR), DisplayBandwidth, DisplayBandwidth (DB), DisplayImportance (DI), DispalyObtrusiveness (DO), ControlScope (CS),

ControlResolution (CR), ControlBandwidth (CB), and/or ControlImportance (CI) that describe the characteristics of the presentation element. Similarly, the information requirement of this presentation element may also be

5 assigned a set of values such as DisplayScope (DS), DisplayResolution (DR), DisplayBandwidth, DisplayBandwidth (DB), DisplayImportance (DI), DispalyObtrusiveness (DO), ControlScope (CS), ControlResolution (CR), ControlBandwidth (CB), and/or

10 ControlImportance (CI) that describe the characteristics of the presentation element. Some presentation elements are essentially only displays, showing information, (hence the use of the word "display"). Other presentation elements allow for inputs or manipulations

15 (hence the use of the word "control"). Still other presentation elements might be both a display and a control. In any case, a set of values DS-CI need to be assigned to the presentation elements so that the reasoner of Figures 3A and 3B can reason. These values

20 may be compared, and a score can be generated that indicates how closely the values of the information requirement and the presentation element match. The block 46 uses these scores for the corresponding presentations. It is also possible to generate a similar

score for each interaction delivery device that indicates how closely the characteristics of the interaction delivery devices match the characteristics stored in the user models 24 and the tasks models 26. The block 46 may 5 be arranged to combine the presentation element score and the interaction delivery device score to form a composite score for the corresponding presentation.

A block 48 then sorts the presentations by score. For each interaction requirement of the user 10 interface being designed as defined by the domain model 22, the user models 24, the task models 26, and the device models 28, a block 50 selects the presentations having the best scores. For each such interaction requirement, a block 52 chooses a presentation off of the 15 best score list. If the designer is not satisfied with the collection of presentations as indicated by a block 54, the designer may change the characteristics and/or requirements of one or more of the models at a block 56 until the designer is satisfied with the resulting user 20 interface. Alternatively or additionally, the designer may cause the interaction design system 12 to choose a different combination of presentations at the block 52. Accordingly, the design of a user interface is an 25 iterative process between the designer and the

interaction design system 12, and this process continues until the designer is satisfied with the designed user interface.

A block 58 generates the presentations as an XML file, and a block 60 applies the interaction delivery devices' XSL to the XML generated by the block 58. It is noted that the result of the block 60 does not include the necessary application-specific communications between the chosen interaction delivery device and the application for which the user interface is being designed. Rather, the result of the block 60 is the interaction appropriate for the interaction delivery device once an appropriate application makes the user interface available to the user.

The device models 28 and the presentation elements library 30 may remain unchanged from user interface design to user interface design. On the other hand, the designer may change the device models 28 and/or the presentation elements library 30 in preparation for the design of one or more user interfaces. The domain model 22, the user models 24, and/or the task models 26 will likely, although not necessarily, change from design to design. One of the advantages of the interaction design system 12 is that it can adapt to new domains, to

new applications, to new tasks, to new interaction delivery devices, to new presentation elements, and to new user requirements simply by storing the appropriate information in the domain model 22, the user models 24, 5 the task models 26, the device models 28, and/or the presentation elements library 30.

Certain modifications of the present invention have been described above. Other modifications will occur to those practicing in the art of the present 10 invention. For example, although the present invention is specifically described above in terms of designing and generating user interfaces, the present invention may be adapted to aiding a designer in the design and generation of products, systems, machines, and arrangements other 15 than user interfaces.

Also, the flow chart of Figures 3A and 3B is described above (particularly with respect to blocks 40-56) as generating all presentations for all interaction requirements prior to generation of the XML file. 20 Instead, a presentation can be added to the XML file one interaction requirement at a time until all interaction requirements are designed into the user interface.

Accordingly, the description of the present invention is to be construed as illustrative only and is

for the purpose of teaching those skilled in the art the best mode of carrying out the invention. The details may be varied substantially without departing from the spirit of the invention, and the exclusive use of all 5 modifications which are within the scope of the appended claims is reserved.